Nonlinear Studies
Vol. 11, No. 2, pp. 173-198 , 2004

Nonlinear Studies

Daytona Beach, FL 2004

Computer-Aided Synthesis of Nonlinear Autopilots for Missiles

P.K. Menon[1], E. J. Ohlmeyer[2]

[1] Optimal Synthesis Inc.
868 San Antonio Road
Palo Alto, CA 94303-4622,U.S.A
e-mail: menon@optisyn.com

[2] Naval Surface Warfare Center
Dahlgren, VA 22448, U.S.A

Abstract Powerful nonlinear approaches for missile autopilot design have recently emerged in the literature, which have the potential to deliver improved missile performance. However, the lack of computational methods has made it difficult for the practicing engineers to exploit these techniques in routine applications. Another factor that has slowed their application is that the missile models are generally available in the form of simulations, rather than as compact set of differential-algebraic equations. This paper discusses five different approaches for computer-aided nonlinear control system design that ameliorate these difficulties. Since these design techniques are based on simulation models, they enable direct synthesis of nonlinear autopilots using missile models of arbitrary complexity. Airframe stabilization of a nonlinear, longitudinal missile model is used to illustrate the design techniques.

1 Introduction

Methods for nonlinear control system design have been of significant interest in the recent literature [1 – 19]. By enabling the design of missile autopilots without employing Taylor series linearization and subsequent gain scheduling, these methods have the potential to enhance the missile performance. While some of these techniques have advanced to a point where they can be routinely employed, the nonlinear design processes are largely based on algebraic manipulations of the underlying mathematical model of the system to be controlled. Although controller design using algebraic manipulations are effective in simpler problems, it becomes increasingly onerous to employ them in practical situations where the

Keywords: nonlinear control, numerical methods, computer-aided design, simulation-based design, missiles, autopilots.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2004		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Computer-Aided Synthesis of Nonlinear Autopilots for Missiles	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Surface Warfare Center,Dahlgren,VA,22448**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**26**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

missile model may contain complex nonlinearities that may not be describable in terms of symbolic expressions, such as sensor-actuator nonlinearities and lookup tables.

In order to motivate subsequent discussions, the flight control system of a typical homing missile is illustrated in Figure 1. The state estimator, the guidance law and the autopilot form the three major components of the system. The state estimator uses measurements from onboard sensors together with a set of missile-target mathematical models to generate estimates of the missile and the target states. The guidance law uses the missile and target state estimates to generate commands for the autopilot. The guidance commands are typically in the form of lateral acceleration components a_y, a_zthat will achieve target interception.

The autopilot has the responsibility for generating actuator inputs for tracking the guidance commands while stabilizing the missile airframe. In addition to the identification of the three flight control subsystems, Figure 1 also illustrates the sets of variables employed by each of them. For instance, the state estimator uses the missile kinematics, an assumed target maneuver model and measurements from the seeker to estimate the target states. The seeker measurements may consist of line of sight rates $\dot{\sigma}$, line of sight angles σ, range $R$and range rate $\dot{R}$. Depending upon the assumed model of the target, the state estimates may include target relative position, velocity and acceleration vectors. The guidance law uses estimated target states and the missile position and velocity vectors to generate lateral acceleration commands for the autopilot. The autopilot tracks these commands while ensuring the stability of the missile short period dynamics [20] consisting of the angle of attack α, angle of sideslip β, pitch rate $q$and yaw body rate $r$states. The roll rate $p$is often regulated about zero.

Traditional approach to autopilot design is to first linearize the missile short period dynamics about an operating condition, and then to apply linear control theory to synthesize a feedback controller. This process is repeated at multiple operating conditions and the controllers are then scheduled with respect to the flight conditions. Typical scheduling variables include Mach number, dynamic pressure, angle of attack and angle of sideslip. In some missiles capable of operating over a wide range of altitudes, the controllers may also be scheduled with respect to altitude. While linear control theory provides elegant algorithms for controller synthesis, the gain scheduling step is largely a trial and error process. Often, the gain scheduling process can consume a significant portion of the design effort. Moreover, the stability and performance guarantees provided by the linear design techniques are generally diluted by the gain scheduling process.

Nonlinear control system design methods seek to eliminate the gain scheduling process, without compromising the performance and stability properties of the closed-loop system. Moreover, these techniques provide a natural setting for including nonlinearities such as actuator saturation in the design process. The objective of the present paper is to advance a set of nonlinear control system design techniques that allow the direct use of numerical simulation models of the missile for nonlinear autopilot synthesis. The analyst exercises control over the design processes through the selection of structural properties of the controller and the parameters that govern a specific design technique. Design techniques presented here can handle a large class of system nonlinearities found in missile autopilot

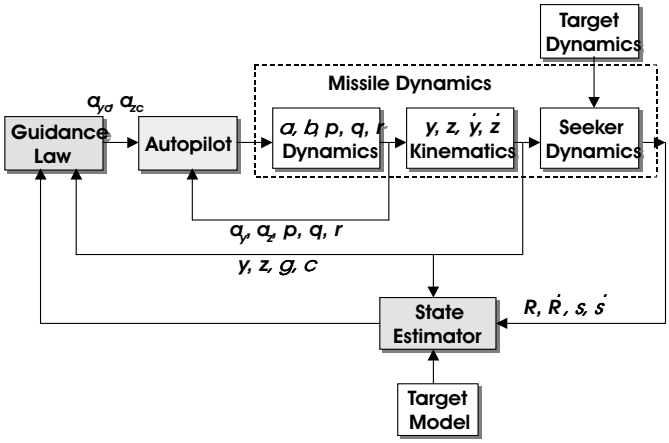

Figure 1: Homing Missile Flight Control System

design problems, including saturation limits, coulomb friction and backlash.

Nonlinear control system design methods will be described in Section 2. These techniques will then be used to illustrate autopilot design using a nonlinear missile model in Section 3. Conclusions will be given in Section 4.

2 Nonlinear Autopilot Design Methods

The nonlinear dynamic models used for autopilot synthesis are assumed to be of specified in the form:

$$\dot{x} = f(x) + g(x)\,u \tag{1}$$

Here, x is the state vector, u is the control vector, $f(x)$ is a vector of state-dependent nonlinear functions and $g(x)$ is a matrix of state-dependent nonlinear functions. The state vector components for autopilot design typically consist of angle of attack α, angle of sideslip β, and p, q, r, the pitch, yaw and roll body rates. Note that the nonlinear functions in equation (1) could be in the form of lookup tables. Every nonlinear control design technique discussed in this paper assumes that a model of this form is available. This model is termed as the *Design Model* in the following. Although this model may include nonlinearities, it should not include sensor or actuator dynamics in order to ensure that their internal states are not used in autopilot computations.

In most missiles, the dynamics may contain input nonlinearities such as saturation and deadzone. In this case, the model may actually be given in the form:

$$\dot{x} = h(x, u) \tag{2}$$

Note that the control variables appear nonlinearly in expression (2). In order to facilitate design using the methods described in this paper, the model in (2) must be transformed into the form of equation (1). This can be accomplished by introducing a new set of control variable u_c, connected to the system through a set

of input dynamic compensators. The input compensators can be of any desired form, as long as the new control variables appear linearly on the right hand side. For instance, input compensators can be in the form of pure integrators, such that:

$$\dot{x} = h(x, u), \dot{u} = u_c \tag{3}$$

Note that the augmented model(3) is now in the standard form with u_c as the new control vector. In addition to allowing transforming the missile model into the standard form, the designer can choose the dynamic properties of the input compensator to shape the frequency content of the signals provided to the actuators. Selection of the input compensator can be thought of as another degree of freedom available to the in the autopilot designer.

Nonlinear autopilot design techniques discussed in this paper may be broadly classified into Transformation Based Methods and Direct Methods. This classification is based on the way the nonlinear design techniques utilize the system dynamic model. In the transformation-based approaches, the given dynamic model is first transformed either to the Brunovsky canonical form [1, 3, 21]or to the state-dependent coefficient form [7 - 9]. Transformed models are then used to design a feedback-linearized autopilot or a state-dependent Riccati equation autopilot.

Direct Methods, on the other hand, do not require any transformation of the given nonlinear system model. These methods employ the user-supplied models in the standard form to synthesize the controllers. The three direct design techniques discussed in this paper are: a) Quickest Descent method [4], b) Recursive Back-Stepping technique [5] and c) Predictive Control [6]approach. The following subsections will describe these autopilot design techniques in further detail.

2.1 Feedback Linearization Method

Feedback linearization techniques have been used for flight control system design for over two decades. This technique has been used extensively in high-performance aircraft flight control system design [10 - 15] and for missile autopilot design [16 - 19]. In order to motivate numerical approach to the feedback linearization method, the following will provide a brief outline of the technique. Additional details on missile autopilot design process using the feedback linearization methodology can be found in References 16 and 17.

The first step in the feedback linearization approach is that of transforming the system dynamics into the Brunovsky [21] canonical form. In this form, the dynamic system under consideration is in the form of decoupled chains of integrators. The transformation process "pushes" all the nonlinearities in the system to the inputs, thereby enabling the construction of an invertible, state-dependent linearizing map. Next, new pseudo-control variables are defined as the product of the state-dependent linearizing map and the actual control variables. The resulting dynamic system is in linear, time invariant form with respect to pseudo-control variables. If the pseudo-control variables are known, actual control variables can be computed using the inverse of the linearizing map. Since the state variables for computing the inverse transform are obtained from feedback, this process is sometimes referred to as global linearization using feedback or feedback linearization.

Feedback linearization process involves the selection of a set of "leading states", which are then repeatedly differentiated until the control variables appear on the right hand sides. In the missile autopilot design process, the angle of attack α and

the angle of sideslip$\beta$are often used as the leading states. Differential equations for these states are then repeatedly differentiated until the fin deflection appears on the right hand sides. For instance, the fin deflection will appear in the second derivative of the angle of attack:

$$\ddot{\alpha} = f_1(\alpha, \beta, p, q, r) + g_{11}(\alpha, \beta)\delta_p + g_{12}(\alpha, \beta)\delta_q + g_{13}(\alpha, \beta)\delta_r = v_1 \tag{4}$$

Here, $\delta_p, \delta_q, \delta_r$are the roll, pitch, and yaw fin deflections. State-dependent nonlinear functions $f_1, g_{11}, g_{12}, g_{13}$ generally include the partial derivatives of aerodynamic forces with respect to angle of attack, angle of sideslip and the body rates. At any given values of the state vector, these functions can be computed numerically using forward difference [22].

Direct force contributions from fin deflections are generally neglected during this transformation process. This is based on the fact that the lateral acceleration components are generated primarily by changing the angle of attack and angle of sideslip using pitch rate and yaw rate. Pitch and yaw rates can be influenced through the moments generated by the fin deflection. Note that in tail-controlled missiles, direct forces generated by fin deflections cause the well-known non-minimum phase response of the autopilot.

An expression for angle of sideslip can also be similarly derived.

$$\ddot{\beta} = f_2(\alpha, \beta, p, q, r) + g_{21}(\alpha, \beta)\delta_p + g_{22}(\alpha, \beta)\delta_q + g_{23}(\alpha, \beta)\delta_r = v_2 \tag{5}$$

The feedback linearization process is analogous to the transformation of linear dynamic systems into the controllable canonical form [21]. Finally, the differential equation for roll rate may be given as:

$$\dot{p} = f_3(\alpha, \beta, p, q, r) + g_{31}(\alpha, \beta)\delta_p + g_{32}(\alpha, \beta)\delta_q + g_{33}(\alpha, \beta)\delta_r = v_3 \tag{6}$$

The variables v_1, v_2, v_3 on the right hand sides of the expressions (4), (5), (6) are the pseudo-control variables in the pitch, yaw and roll axes. Note that the system is linear with respect to the pseudo-control variables. In the interests of simplifying notation, Mach number and altitude dependencies of the state-dependent nonlinear functions on the right hand sides of the expressions (4) – (6) have been dropped. However, the present development can be used without any change to include that case also.

After the system is transformed into the feedback-linearized form, any linear control design method can be applied to derive the pseudo control variables. This step is particularly simple because the feedback linearized dynamic system is in the form of decoupled chains of integrators. Techniques such as pole placement [23], LQR [24] and sliding mode control [2] can be directly applied to derive feedback control laws for pseudo-control variables. For instance, using either the pole placement method or the LQR method, the pseudo-control laws for regulating the missile dynamics can be found in the form:

$$v_1 = K_1\ \alpha + K_2\ \dot{\alpha} \tag{7}$$

$$v_2 = K_3\,\beta + K_4\,\dot{\beta} \tag{8}$$

$$v_3 = K_5\,p \tag{9}$$

Once the pseudo-control vector is available, corresponding fin deflections can be computed using the expressions (4), (5), (6) as:

$$\begin{bmatrix} \delta_p \\ \delta_q \\ \delta_r \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix}^{-1} \begin{bmatrix} K_1\,\alpha + K_2\,\dot{\alpha} - f_1 \\ K_3\,\beta + K_4\,\dot{\beta} - f_2 \\ K_5\,p - f_3 \end{bmatrix} \tag{10}$$

The foregoing approach can be modified to permit lateral acceleration command tracking by augmenting the system with integral tracking error states. In this case, two new integral error states are defined, with the corresponding differential equations:

$$\dot{e}_z = a_{zc} - h_1(\alpha, \beta, \delta_p, \delta_q, \delta_r) \tag{11}$$

$$\dot{e}_y = a_{yc} - h_2(\alpha, \beta, \delta_p, \delta_q, \delta_r) \tag{12}$$

The nonlinear functions h_1 and h_2 relate the angle of attack, angle of sideslip and the fin deflections to the lateral acceleration components. Expressions (11) and (12) can be used in conjunction with the angle of attack, angle of sideslip and body rate dynamics to derive the feedback linearizing transformations. As in the case of regulator design, the effect of fin deflections on the lateral acceleration components will be neglected during the transformation process.

From the foregoing discussions, it can be observed that the numerical implementation of the feedback linearization technique requires the identification of the leading states, their relationship to the control variables, and the right-hand-sides of the system dynamics. This information can be used by a numerical differencing scheme to find the partial derivatives required for computing the feedback linearizing transformations. For the missile autopilot design problem, the leading states and their dominant relationships to the control variables can be symbolically expressed as:

$$\delta_p \rightarrow p \tag{13}$$

$$\delta_q \rightarrow q \rightarrow \alpha \rightarrow e_z \tag{14}$$

$$\delta_r \rightarrow r \rightarrow \beta \rightarrow e_y \tag{15}$$

Note that these expressions only capture the dominant relationships. Coupling terms in the transformation can be computed by determining the dependence of each of the state variables on other state variables. Numbering the states $e_y, e_z, \alpha, \beta, p, q, r$ sequentially, and the control variables $\delta_p, \delta_q, \delta_r$ sequentially, the symbolic relationships in (13, (14), (15) can be captured in the form of a matrix:

$$\begin{bmatrix} 5 & 0 & 0 \\ 6 & 3 & 2 \\ 7 & 4 & 1 \end{bmatrix} \tag{16}$$

Each row of this matrix corresponds to a control variable, and each column corresponds to a state. For instance, the first row suggests that the roll fin deflection influences the roll rate. Similarly, the second row suggests that the pitch fin deflection mainly influences the pitch rate, which in turn influences the angle of attack. The angle of attack strongly influences the lateral acceleration component in the pitch plane. The matrix (16), together with the right-hand-sides of the

state equations can used to configure a numerical differencing procedure to automatically construct feedback linearizing transforms at any given value of the state vector.

As discussed elsewhere in this paper, for a nonlinear control technique to be useful in applications, it should be able to directly employ a simulation model of the dynamic system. By insisting that the simulation model be arranged such that the integrators are included in a separate block as shown in Figure 2, right-hand-sides of the state equations can be readily isolated for the numerical feedback linearization process.

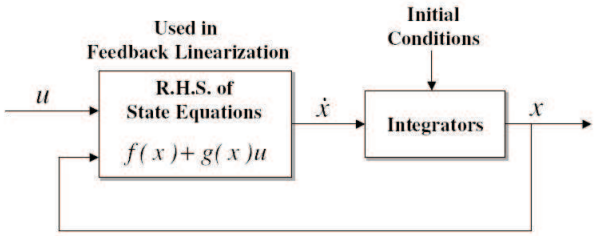

Figure 2: Desired Form of the Simulation Model

A software package is currently available [25, 26], that uses a simulation model of the form shown in Figure 2 to derive feedback linearizing transforms, and the corresponding inverse transforms. Numerical algorithms used in this software were developed over the past decade, and have been reported in References 13, 27 - 29. Once the model is feedback linearized, numerical methods for linear control system design [22] can be employed to design the pseudo-control laws.

It has previously been observed [11] that in higher-order dynamic systems such as the autopilot design problem, the nonlinear controller robustness can be significantly enhanced by designing multiple time-scale controllers. Robustness in time-scale separated controllers result from the fact that higher-order partial derivatives of the nonlinearties on the right hand sides of the nonlinear model are not used in control law derivation. Additionally, time-scale separated nonlinear controllers can exploit the hierarchical structure of the system states to simplify control law implementation. Advantages of time scale separation in the context of missile autopilot design have been previously investigated [17]. The feedback linearization procedure outlined in the foregoing can be readily adapted to allow for time scale separation of the system dynamics. In this case, the user will have the additional responsibility for identifying the state variables to be used in slow and fast time-scales.

2.2 State-Dependent Riccati Equation Method

State Dependent Riccati Equation (SDRE) method [7 - 9] is another technique that uses transformed dynamic model for nonlinear controller design. By defining the nonlinear dynamics of the system in terms of state dependent matrices, this technique allows the derivation of nonlinear controllers using techniques similar to that of the LQR [24] technique. The first step in the SDRE technique is the

transformation of the user specified dynamic model given in the standard form $\dot{x} = f(x) + g(x)$ u into the State Dependent Coefficient (SDC) form [7, 9].

$$\dot{x} = A(x)\ x + g(x)\ \mathrm{u} \tag{17}$$

The matrix $A(x)$ is an instantaneous parameterization of the state-dependent nonlinear functions $f(x)$. For multivariable systems, infinite number of such realizations can be shown to exist [7, 9]. However, only those parameterizations for which the pair $[A(x)\,,\ \mathrm{g(x)}]$ is controllable at the given x should be considered for the design.

Note that the SDC parameterization is distinct from the conventional Taylor series linearization. Given the simulation model of a dynamic system in the form illustrated in Figure 2, instantaneous SDC parameterization can be obtained by evaluating the vector nonlinear function $f(x)$ using a set of linearly independent *probe* vectors ζ_2, ..., ζ_n. As a practical matter, since the behavior of the nonlinearities in the neighborhood of the current system state are not explicitly known, it is wise to choose probe vectors that are close to the current state vector. At the current state vector, the vector nonlinear function $f(x)$ can be extracted from the simulation model by setting the control vector to zero.

The probe vectors are constructed by adding small magnitude perturbation vectors $\sigma_{\mathrm{y}}\mathrm{sg}\sigma_{\mathrm{z}}\mathrm{sg}\sigma_4\mathrm{suuuu}\sigma_n\mathrm{sgoghe}$ nominal state vector to yield a set of linearly independent vectors:

$$\zeta_2 = x + \sigma_2,\ \zeta_2 = x + \sigma_3,\ \zeta_3 = x + \sigma_4,\ .....,\ \zeta_n = x + \sigma_n \tag{18}$$

The nonlinear function $f(x)$ is next evaluated using these linearly independent vectors to assemble a matrix equation of the form:

$$\left[\ f(x) \quad f(\zeta_2) \quad ..... \quad f(\zeta_n)\ \right] = A(x)\ \left[\ x \quad \zeta_2 \quad ..... \quad \zeta_n\ \right] \tag{19}$$

At any given value of x, this linear matrix equation can be solved for the elements of $A(x)$. Since the probe vectors and the state vector are linearly independent, this equation is well conditioned, and can be solved using well-known numerical linear algebraic algorithms.

Note that the foregoing computations will have to be carried out at every sample. The SDC matrix $A(x)$ from these computations can next be used to formulate and solve the SDRE control problem.

As an aside, it is interesting to examine the relationship between the numerical construction of the SDC model and the conventional Taylor series linearization. If the perturbation vectors $\sigma_{\mathrm{y}}\mathrm{sg}\sigma_{\mathrm{z}}\mathrm{sg}\sigma_4\mathrm{suuuu}\sigma_n\mathrm{sgre}$ small, it can be found that:

$$A \cong \frac{\partial f}{\partial x},\ \ at\ x = 0 \tag{20}$$

Note that this corresponds to the Taylor series linearization of the system dynamics about the origin. Thus, the present numerical methodology for constructing the SDC model automatically reverts to Taylor series linearization of the system dynamics near the origin of the state space. For constant control influence matrix case, the present SDC parameterization scheme preserves the controllability

properties of the dynamic system near the origin. Since the only restriction on the probe vectors is that they be linearly independent, it is possible to construct an infinite variety of SDC parameterizations for a given dynamic system.

The SDC form of the system state equations is next used to cast the control problem as an infinite-horizon nonlinear regulator minimizing the cost:

$$J = \frac{1}{2}\int_{t_0}^{\infty} x^T Q\left(x\right) \mathrm{x} + \mathrm{u}^{\mathrm{T}} R(x) \mathrm{u}\ \mathrm{dt} \quad (21)$$

subject to the nonlinear differential constraint: $\dot{x} = A\left(x\right)\ x + g\left(x\right)\ \mathrm{u}$. The state-dependent matrices $Q(x)$ and $R(x)$ are chosen by the designer to achieve desired properties of the closed-loop system.

References 7 and 9 show that the solution to this problem can be obtained by solving a state dependent Riccati equation:

$$A^T\left(x\right) P\left(x\right) + P\left(x\right) A\left(x\right) - P\left(x\right) g\left(x\right) R^{-1}\left(x\right) g^T\left(x\right) P\left(x\right) + Q = 0 \quad (22)$$

The state dependent feedback gain can then be computed as:

$$K\left(x\right) = R^{-1}\left(x\right) g^T\left(x\right) P\left(x\right) \quad (23)$$

The SDRE nonlinear control law is of the form:

$$u = -K\left(x\right)\ x \quad (24)$$

Reference 7 has shown that under rather mild restrictions on $Q(x)$ and $R(x)$, the SDRE control law will globally stabilize the nonlinear dynamic system. Figure 3 illustrates the computational steps involved in the SDRE technique. At each time step, SDC parameterization of the dynamic system is generated and used to formulate the algebraic Riccati equation. This equation is then solved for the state-dependent feedback gains.

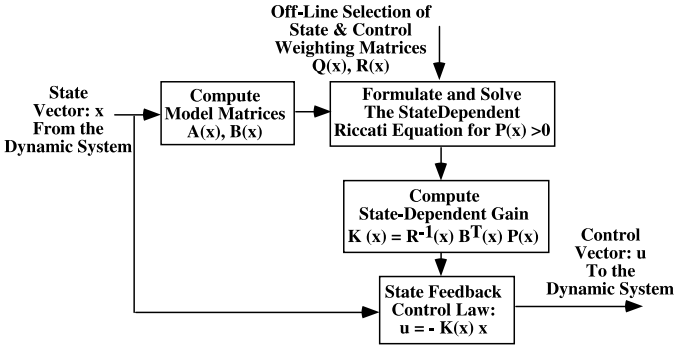

Figure 3: SDRE Control Computations

Note that the SDRE technique is computationally demanding, requiring the solution of a 7×7 algebraic Riccati equation at each sample. However, previous

research [30] has demonstrated that these computations are well within the capabilities of commercially available processors.

2.3 Quickest Descent Method

The Quickest Descent method discussed in Reference 4 is a Lyapunov function optimizing approach to nonlinear feedback controller design. This is perhaps the simplest of all the five nonlinear control techniques discussed in this paper. In this approach, the control problem is viewed as a function minimization problem in the state space. A descent function, $W(x)$ satisfying certain specified properties is first selected. Control vector is then chosen to minimize this descent function.

The descent function $W(x)$ is required to be bounded, continuous and continuously differentiable in the region of interest. In addition, the target state is required to be contained within the region of interest. Note that these requirements are more restrictive than the choice of Lyapunov functions. Although no general guidelines are available for the selection of descent functions, it appears that physical quantities such as the total energy in the system can be used as the starting point. In problems such missile autopilot design, a properly defined quadratic function of the states can be used as the descent function.

Once the descent function $W(x)$ is selected, the feedback control $u(x)$ is chosen so that $W(x)$ decreases at each state of the system. If the minimization process is cast as a steepest decent optimization problem, the resulting technique can be termed as the steepest decent control methodology. Reference 4 shows that a more direct approach is to choose the control variables to minimize the time-rate of change of the descent function approach. In this case, the control methodology can be termed as the *Quickest Decent* method.

In the quickest descent method, control is obtained by minimizing the time rate of change of the descent function. Thus, the optimization problem is of the form:

$$\min_u \frac{dW(x)}{dt} \quad \text{or} \quad \min_u \left[\frac{\partial W}{\partial x} \{ f(x) + g(x) \ u \} \right] \tag{25}$$

Since the control variable appears linearly in the system dynamics, the optimization problem is meaningful only if the control variables are constrained. The control constraints can be specified in the form: $|u| \leq u_{\max}$. The quickest descent control is then given as:

$$If \ \left\{ \frac{\partial W}{\partial x} g(x) \right\} > 0, \ u = u_{\min} \tag{26}$$

$$If \ \left\{ \frac{\partial W}{\partial x} g(x) \right\} < 0, \ u = u_{\max} \tag{27}$$

Note that the control is bang-bang. Under the present formulation of the quickest descent method, the control variables will chatter between their limits as the system approaches the minimum of the descent function $W(x)$.

In missile autopilot design problem, the descent function can often be specified

as a quadratic function of the form:

$$W(x) = \begin{bmatrix} e_z & e_y & \alpha & \beta & p & q & r \end{bmatrix} P \begin{bmatrix} e_z \\ e_y \\ \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix} \tag{28}$$

with P being a positive definite matrix. The elements of the matrix P has to be carefully to ensure that the control objectives are satisfied. For instance, diagonal elements of P can be chosen to drive state vector components such as the integral tracking error states and the body rates to zero, while the off-diagonal elements can be chosen to preserve the coupling between the missile state variables to achieve the desired response. Selection of the control bounds and the matrix P are the two degrees of freedom available to the designer in the quickest descent control technique.

2.4 Recursive Back-stepping Method

As can be observed from Section 2.1, the feedback linearization method cancels the system nonlinearities and replaces them with a linear dynamic system. The main premise behind the recursive back stepping technique [5] is that certain portions of the system nonlinearities are worth preserving. This objective is satisfied by formulating the control problem using the second method of Lyapunov. However, since there is no direct approach for constructing Lyapunov functions for multivariable nonlinear dynamic systems, the backstepping procedure relies on a recursive procedure. Just as the system nonlinearities were pushed-back to the inputs in the feedback linearization methodology, recursive back stepping technique constructs Lyapunov function for the nonlinear dynamic system by stepping-back from the output state variables to the controls. In some respects, this technique bears a strong resemblance to the multiple time-scale feedback linearization design technique.

The recursive back stepping design technique assumes that the model is specified in a triangular form as shown in equations (29) – (31):

$$\dot{x}_1 = f_1(x_1) + g_1(x_1)\ x_2 \tag{29}$$

$$\dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)\ x_3 \tag{30}$$

$$\vdots$$

$$\dot{x}_n = f_n(x_1, x_2, \cdots, x_n) + g_n(x_1, x_2, \cdots, x_n)\ u \tag{31}$$

Here, $x_1, x_2, \ldots x_n$are the components of the state vector. Each scalar system is stabilized with the following state as the control variable. For example, $x_2$serves as the control variable for x_1- dynamics, x_3 for x_2- dynamics and so on. Note that such a structure can be found in missile dynamics, if the direct force contributions arising from fin deflections are neglected. The triangular structure in missile dynamics consists of the pitch and yaw rates generating angle of attack and angle of sideslip, which in turn result in the lateral acceleration components.

Controllers are synthesized for each scalar dynamic system using the second method of Lyapunov. At each stage of the backstepping process, Lyapunov functions are selected to preserve desired nonlinearities. In most practical problems, quadratic Lyapunov functions can be used to realize most of the benefits of this method.

In the missile autopilot example, the back stepping process begins with the integral tracking error states e_z, e_y and the roll rate p. Control system design for the roll channel does not involve any backstepping, since the roll fin deflection is directly related to the roll acceleration. A quadratic Lyapunov function of the form: $\frac{1}{2}p^2$can be used to derive the roll fin deflection as a function of the missile states that will drive the roll rate to zero.

In the pitch and yaw channels, angle of attack and angle of sideslip form the control variables for the integral tracking error states in the first stage of the back stepping process. A Lyapunov function of the form: $\frac{1}{2}\left[e_z^2 + e_y^2\right]$ can be used to derive the angle of attack α_1 and angle of sideslip β_1 that will drive the integral tracking error states to zero. Next, the Lyapunov functions are augmented by quadratic terms in $(\alpha_1$-αp and $(\beta_1$-βp to form the second stage of the backstepping process. In the second stage, the pitch rate q_2 and the yaw rate r_2 are the "control like" variables. These variables are chosen to ensure that the time rate of change of the augmented Lyapunov function will be less than or equal to zero. In the last stage of the back stepping process, the Lyapunov function is augmented by quadratic terms in $(q_2 - q)$ and $(r_2 - r)$. Pitch and yaw fin deflections that drive the time rate of change of the resulting Lyapunov function to be less than or equal to zero. At each stage of the backstepping process, the intermediate states are eliminated using the given state equations, see Reference 5 for details.

In numerical implementations of the backstepping process, the user can specify the backstepping sequence through a matrix similar to that used for specifying the feedback linearization, expression (15). The user can also specify the form of the Lyapunov functions to be used in each step of the backstepping procedure. Since the nonlinearities contained in the missile dynamics is generally limited to lookup tables, products of the state variables and transcendental functions, quadratic Lyapunov functions are often adequate to obtain good response from the autopilot. This information together with a simulation model of the form shown in Figure 2 are sufficient to develop a numerical procedure for recursive backstepping process.

2.5 Predictive Control

The predictive control methodology [6] has been popular in the process control industry for the past two decades. In this technique, a control history that will drive the system states to the desired values at the end of a prediction interval is computed at every sample interval using an optimization algorithm. Most of the techniques described in the literature employ multi-step predictors to implement the controllers. In nonlinear systems, the use of predictive control technique requires the use of an on-line iterative optimization algorithm. Since the convergence of optimization techniques for general nonlinear problems is not assured, the performance of the predictive control technique cannot be guaranteed in these situations

However, if the nonlinear dynamic system is of the standard form given by equation (1), and if the control problem is cast as a one-step-ahead predictive

control, the system performance becomes more predictable. This approach is adopted in the present formulation of the nonlinear predictive control technique. Thus, the control problem is cast as the minimization of the quadratic objective function

$$J \triangleq \mathrm{x}_{i+1}^T \ \mathrm{Q\,(x,t)} \ \mathrm{x}_{i+1} + \mathrm{u}_i^T \ \mathrm{R\,(x,t)} \ \mathrm{u_{i(32)}}$$

with respect to the control variable u, subject to the differential constraint:

$$\dot{x} = f\,(x) + g\,(x) \ \mathrm{u} \tag{33}$$

The differential constraint can be used to eliminatex_{i+1} from the objective function by defining a numerical integration algorithm. Any one of the several numerical integration techniques can be used for this purpose. However, if the nonlinear dynamic system is specified in the standard form, the linearity of the differential constraints with respect to the control variables can be preserved if techniques such as Euler's integration method or the Adams-Bashforth numerical integration scheme [22] are employed. For instance, if the Euler's integration formula is used to integrate the differential constraint (33) with a step size Δt, the control vector that minimizes (32) can be found to be:

$$u_k = -\left[R^{-1} + \Delta t^2\, g_k^T\, Q\, g_k\right]^{-1} \left[R^{-1} + \Delta t\, g_k^T\, Q\, x_k + \Delta t^2 g_k^T Q\, f_k\right] \tag{34}$$

The use of multi-step integration formulae will produce more complex expressions for one-step-ahead optimal control. The state and control weighting matrices Q and R must be chosen to satisfy the descent property:

$$\mathrm{J}_{i+1} \leq \mathrm{J}_i \tag{35}$$

As in the Quickest Descent method, the structure of the state weighting matrix must be chosen to establish the desired relationships between the state variables.

3 Design Example – Nonlinear Autopilot Design Using Missile Longitudinal Dynamics

The application of the nonlinear control system design methods described in Section 2 will be applied to the missile autopilot design problem in this section. In the interests of a compact presentation, formulation of the nonlinear regulation problem will only be examined. As discussed in Section 2, extension to the command tracking case involves the augmentation of the system dynamics with integral tracking error states, and is direct. Longitudinal dynamic model of a missile used to illustrate the nonlinear autopilot design techniques in this paper is obtained from Reference 8.

The dynamics of a generic homing missile coasting in the vertical plane is given by the expressions (36) through (37). The model incorporates Mach number M, angle of attack α, flight path angle γ and pitch rate q as the state variables and the pitch fin deflection δ as the control variable. The aerodynamic axial force,

normal force and the pitching moment have all been expressed in terms of the state variables on the right hand sides of these differential equations.

$$\begin{aligned}\dot{M} = {} & 0.4008\ M^2\alpha^3 \sin(\alpha) - 0.6419\ \mathrm{M}^{2|\alpha|\alpha \sin(\alpha) - 0.2010}\ \mathrm{M}^{2\left(2-\frac{M}{3}\right)\alpha \sin(\alpha)} \\ & -0.0062\ \mathrm{M}^{2-0.0403}\ \mathrm{M}^{2\sin(\alpha)\ \delta}\text{-}0.0311\ \sin_{(\gamma)}\end{aligned} \tag{36}$$

$$\begin{aligned}\dot{\alpha} = {} & 0.4008\ M\alpha^3 \cos(\alpha) - 0.6419\ \mathrm{M}\,|\alpha|\,\alpha \cos(\alpha) - 0.2010\ \mathrm{M}\left(2-\tfrac{M}{3}\right)\alpha \cos(\alpha) \\ & -0.0403\ \mathrm{M}\cos(\alpha)\ \delta + 0.0311\ \tfrac{\cos(\gamma)}{M} + q\end{aligned} \tag{37}$$

$$\begin{aligned}\dot{\gamma} = {} & -0.4008\ M\alpha^3 \cos(\alpha) + 0.6419\ \mathrm{M}\,|\alpha|\,\alpha \cos(\alpha) + 0.2010\ \mathrm{M}\left(2-\tfrac{M}{3}\right)\alpha \cos(\alpha) \\ & +0.0403\ \mathrm{M}\cos(\alpha)\ \delta\text{-}0.0311\ \tfrac{\cos(\gamma)}{M}\end{aligned} \tag{38}$$

$$\begin{aligned}\dot{q} = {} & 49.82\ M^2\alpha^3 - 78.86\mathrm{M}^{2|\alpha|\alpha + 3.60}\ \mathrm{M}^{2\left(-7-\frac{8M}{3}\right)\alpha - 14.54}\ \mathrm{M}^{2\delta} \\ & \text{-}2.12\mathrm{M}^{2q}\end{aligned} \tag{39}$$

Missile autopilot design problem is concerned with the short-period dynamics with α and q as the state variables. The flight path angle γ and Mach number M, are necessary for calculating the aerodynamic and gravitational forces. The differential equations for γ and M can be treated as auxiliary expressions that contribute to the right hand sides of the short period dynamics. Pitch fin deflection, δ is the control variable. Note that the gravitational acceleration term is normally neglected in missile autopilot design problems.

As the first step in the autopilot design process, a computer simulation of the missile is constructed using the system dynamics. A step size of *10ms* was used in the simulations. The pitch rate is identified as the first state, and the angle of attack is denoted as the second state in the system. Both these states, Mach number and the flight path angle were assumed to be available from measurements. No actuator dynamics was included in the simulation. Design parameters and simulation results for each of the five techniques will be given in the following sections.

3.1 Nonlinear Regulator Designs Using Feedback Linearization

The first step in this design technique is that of generating the feedback linearization map. In the autopilot design example, the feedback linearization sequence can be specified as:

$$\delta \rightarrow q \rightarrow \alpha \tag{40}$$

Since the pitch rate and angle of attack states have been sequentially numbered, this can also be represented by a row vector: [1 2]. This notation implies that the fin deflection, the only control variable in the problem, will be used to generate pitch rate, which would then produce the desired angle of attack. Thus, the feedback linearization map will be constructed by first differentiating the right hand side of the differential equation for angle of attack, followed by a substitution of the right hand side of the pitch rate equation. Forward difference method was used for the numerical computation of the partial derivatives. The state perturbations used for these computations were 10^{-4}Radians.

Feedback linearized form of the system dynamics is:

$$\ddot{\alpha} = v \tag{41}$$

This model can be used to design feedback control laws for the pseudo-control variable. The control law can then be inverse transformed to obtain fin deflections that will regulate the nonlinear dynamic system. This process will be illustrated using three different control approaches.

(a) Pole Placement design:

The poles of the feedback linearized dynamic system were chosen to provide critical damping and a natural frequency of about 14 rad/sec. The response of the nonlinear missile dynamics under this feedback linearized control law is illustrated in Figures 4 and 5. It can be observed that the system responses are essentially that of the feedback linearized system. The responses have been found to remain invariant even under ±10% perturbations in aerodynamic force and moment models.

(b) Linear Quadratic Regulator design:

The LQR approach is used next to design the feedback linearized control law. The state-weighting matrix was chosen to be:

$$\begin{bmatrix} 1000 & 0 \\ 0 & 1 \end{bmatrix} \tag{42}$$

The control weighting was chosen as 0.01.

The angle of attack and pitch rate responses to the initial conditions are illustrated in Figure 6. Corresponding fin deflection time history is given in Figure 7. It may be observed that the responses from the feedback linearized LQR design are comparable to the ones from the pole placement design. As with that case, ±10% perturbations in the system model had no observable effects on the closed-loop system responses.

(c) Sliding Mode design:

Sliding mode control methodology from Reference 2 is next used to design a regulator for the feedback linearized system dynamics. The sliding surface parameter λ is chosen as $--20$, uncertainty parameters are set to zero, the convergence parameter to the sliding surface η is chosen as 20, and the thickness of the boundary layer ε is chosen as 0.1. The time histories of the angle of attack and pitch rate for the feedback linearized sliding-mode autopilot are given in Figure 8. Corresponding fin deflection history is in Figure 9. Note that the response of the closed-loop system is significantly different from that of the pole placement or LQR designs.

3.2 State Dependent Riccati Equation Autopilot Design

State perturbations used for constructing the SDC form of the missile model were 10^{-4}Radians. Although the design technique allows state-dependent state and control weighting matrices, constant weights were used in the present case. These were:

$$Q = \begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix}, R = 10 \tag{43}$$

Figure 4: Angle of Attack and Pitch Rate Responses for the Feedback Linearized Regulator with Pole Placement Design

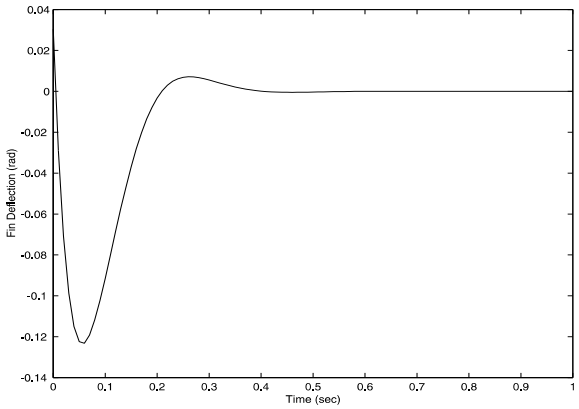

Figure 5: Fin Deflection for the Feedback Linearized Regulator with Pole Placement Design

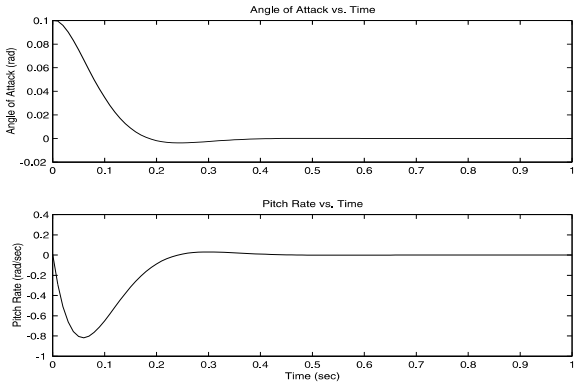

Figure 6: Angle of Attack and Pitch Rate Responses for the Feedback Linearized Regulator with LQR Design

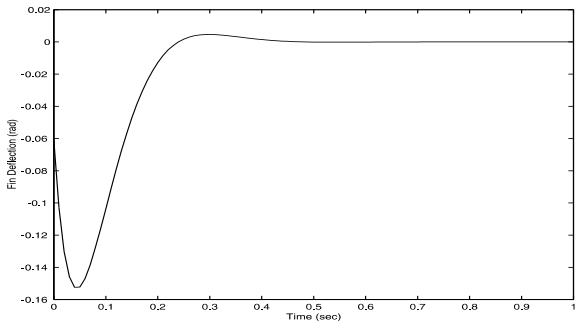

Figure 7: Fin Deflection for the Feedback Linearized Regulator with LQR Design

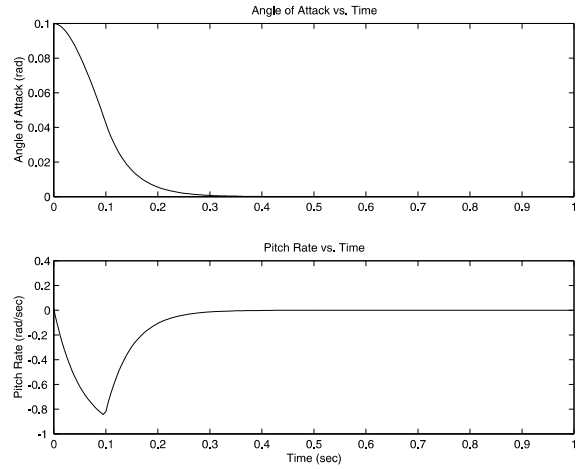

Figure 8: Angle of Attack and Pitch Rate Responses for the Feedback Linearized Regulator with Sliding Mode Design

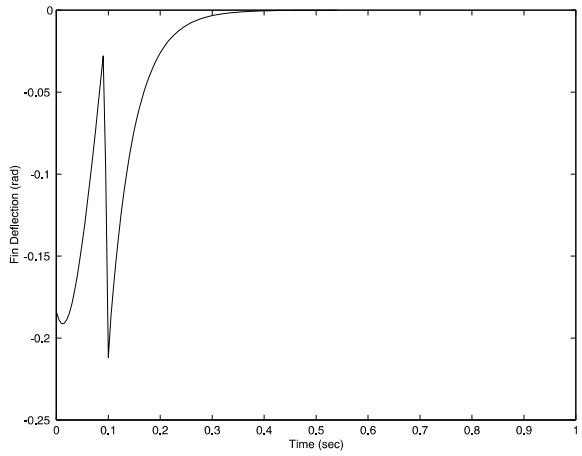

Figure 9: Fin Deflection for the Feedback Linearized Regulator with Sliding Mode Design

Response of the SDRE autopilot are given in Figures 10 and 11. Note that it is possible to further improve the speed of response by choosing alternate design weights.

3.3 Quickest Descent Design

The design parameters for the Quickest Descent technique are the descent function and the limits on the control variables. As discussed elsewhere in this paper, the descent function can be chosen as a quadratic function in most practical design problems. For the autopilot design, the descent function was chosen to be of the form:

$$W(x) = \begin{bmatrix} \alpha & q \end{bmatrix} P \begin{bmatrix} \alpha \\ q \end{bmatrix} \tag{43}$$

The weighting matrix and the control limits were chosen to be:

$$P = \begin{bmatrix} 5000 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \ |u| \ \leq 0.01 \tag{44}$$

The state and control histories obtained from a closed-loop simulation of the quickest descent autopilot are given in Figures 12 and 13. It may be observed that the control chatters after an initial bang-bang region.

3.4 Recursive Backstepping Autopilot Design

Recursive backstepping autopilot is synthesized using the angle of attack and pitch rate as the backstepping variables, and fin deflection as the control variables. Numerical partial derivatives required in the backstepping procedure were computed using forward difference technique [2] with 10^{-4}Radians perturbation in the states.

Quadratic Lyapunov functions were used in each step of the process, with a convergence rate of 20 in each stage. The angle of attack and pitch rate time

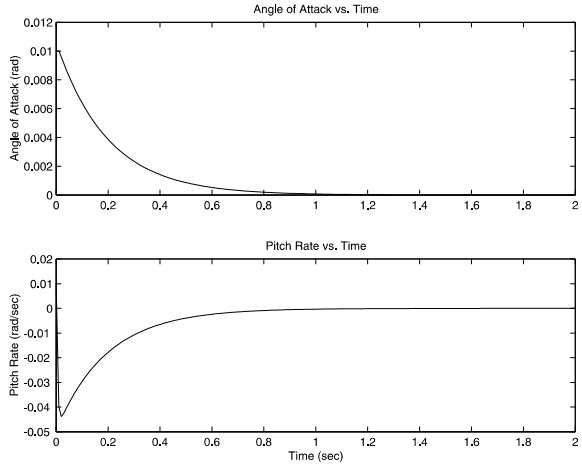

Figure 10: Angle of Attack and Pitch Rate Responses for the Feedback Linearized Regulator with SDRE Design

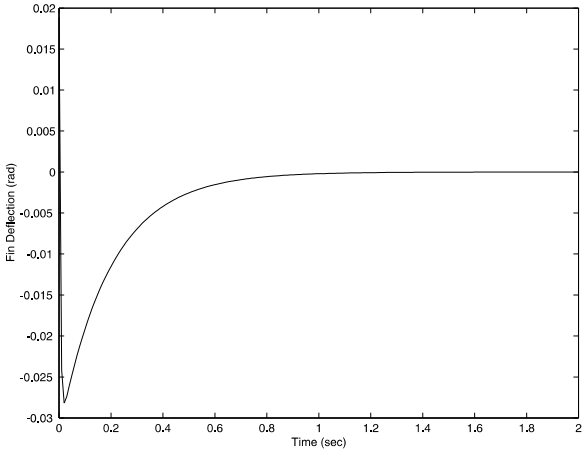

Figure 11: Fin Deflection for the Feedback Linearized Regulator with SDRE Design

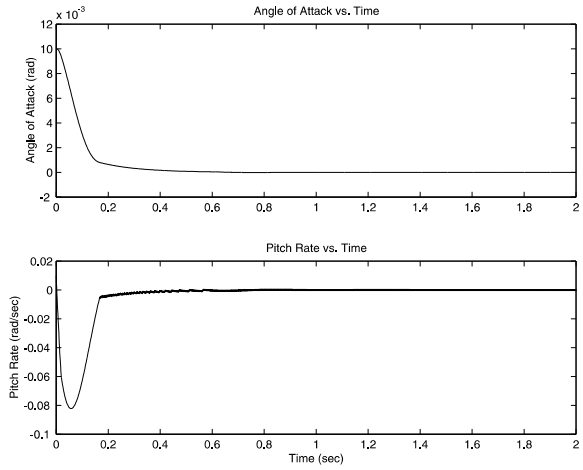

Figure 12: Angle of Attack and Pitch Rate Responses for the Feedback Linearized Regulator with Quickest Descent Design

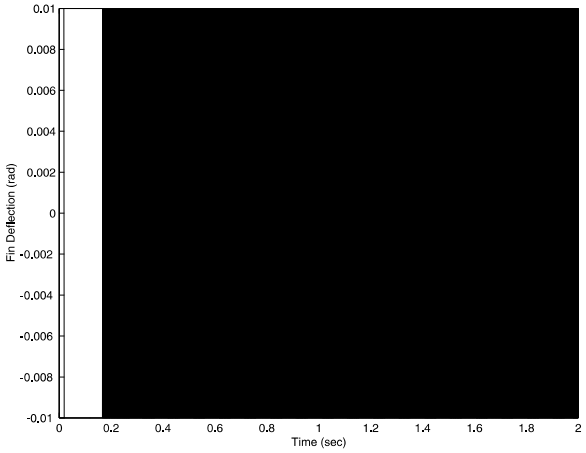

Figure 13: Fin Deflection for the Feedback Linearized Regulator with Quickest Descent Design

histories for the recursive backstepping autopilot are shown in Figure 14. Fin deflection history is given in Figure 15.

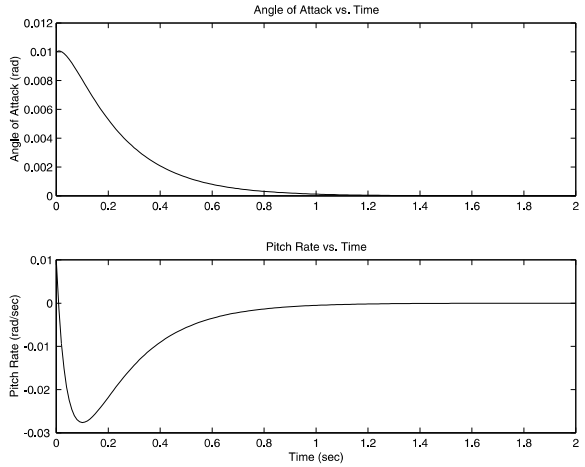

Figure 14: Angle of Attack and Pitch Rate Responses for the Backstepping Autopilot Design

3.5 Predictive Autopilot Design

Predictive autopilot design used a second-order Adams-Bashforth integration algorithm to formulate the predictive performance index. The state weighting matrix and the control weights were chosen as:

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \quad R = 2$$

Note that this design equally weights angle of attack response and the pitch rate response. The state vector and control response of the predictive autopilot are illustrated in Figures 16 and 17. As with other design techniques with quadratic criteria, the speed of response of the predictive autopilot can be improved by reducing the control weighting or by increasing the state weights.

4 Conclusions

This paper discussed five different approaches for computer-aided nonlinear autopilot design. These are numerical implementations of the nonlinear control system design techniques discussed in the literature. Consequently, they can use numerical simulation models of the dynamic systems of arbitrary complexity to automatically derive stable nonlinear closed-loop control systems. The focus of the present paper was on the application of these techniques to the design missile autopilots. General design approaches were first outlined, and illustrated using a nonlinear model of a missile. The approaches presented in this paper have been employed to design full-order autopilots and integrated guidance-control systems using a variety of missile models. The examples presented in this paper show that numerical

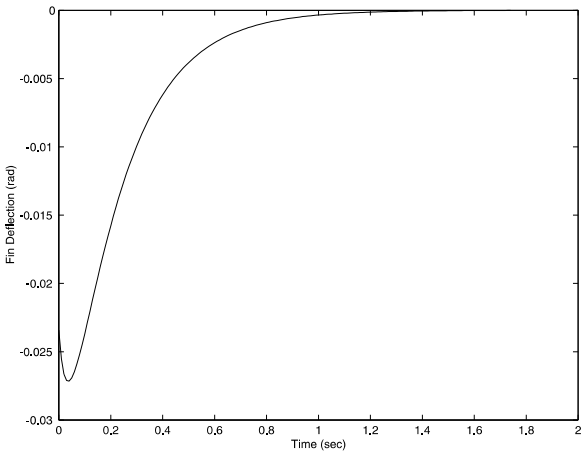

Figure 15: Fin Deflection history for the Backstepping Autopilot Design

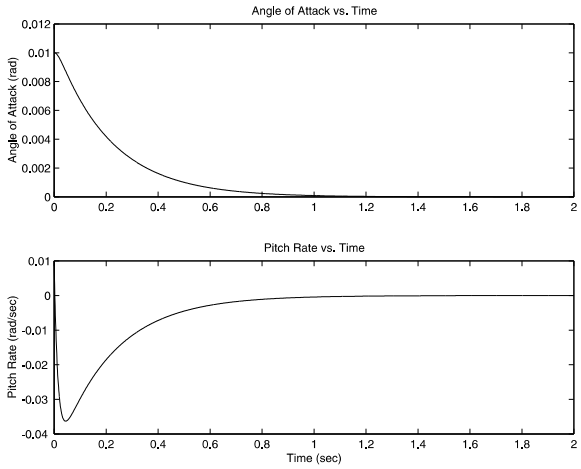

Figure 16: Angle of Attack and Pitch Rate Responses for the Predictive Autopilot Design

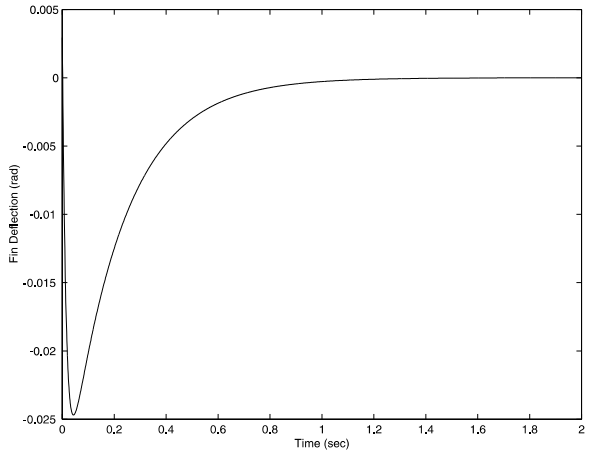

Figure 17: Time history of the Fin Deflection for the Predictive Autopilot Design

approach to nonlinear control system design is feasible, and can be carried out with a level of confidence comparable to that of linear design techniques.

5 Acknowledgement

This work was supported under the US Navy Contract No. N00024-97-C-4178.

References

[1] Isidori, A., Nonlinear Control Systems, Springer-Verlag, 1985, New York, NY.

[2] Slotine, J.J.E. and Li, W., Applied Nonlinear Control, Prentice Hall, 1991, Englewood Cliffs, NJ.

[3] Marino, R., Tomei, P., *Nonlinear Control Design*, Prentice-Hall, 1995, New York, NY.

[4] Vincent, T. L. and Grantham, W. J., *Nonlinear and Optimal Control Systems*, John Wiley, 1997, New York, NY.

[5] Krstic, M., Kanellakopoulos, I. and Kokotovic, P., *Nonlinear and Adaptive Control Design*, John Wiley and Sons, 1995, New York, NY.

[6] *The Control Handbook*, Levine, W. S. (Editor), CRC Press, 1996, Boca Raton, FL.

[7] Cloutier, J. R., D'Souza, C. N. and Mracek, C. P., "Nonlinear regulation and nonlinear H_∞ control via the State Dependent Riccati Equation technique,

Part 1: Theory, Part 2: Examples", *Proceedings of the International Conference on Nonlinear Problems in Aviation and Aerospace*, May 1996, Daytona Beach, FL.

[8] Mracek, C.P. and Cloutier, J. R., "Missile longitudinal autopilot design using the State Dependent Riccati Equation method", *Proceedings of the International Conference on Nonlinear Problems in Aviation and Aerospace*, May 1996, Daytona Beach, FL.

[9] Cloutier, J. R., "State-Dependent Riccati Equation Techniques: An Overview", *American Control Conference*, June 4 - 6, 1997 Albuquerque, NM.

[10] Meyer, G., and Cicolani, L., "Application of Nonlinear Systems Inverses to Automatic Flight Control Design System Concepts and Flight Evaluation," *Theory and Applications of Optimal Control in Aerospace Systems*, AGARDograph 251, P. Kant (Editor), July 1981.

[11] Menon, P. K., Badgett, M. E., Walker, R. A., and Duke, E. L., "Nonlinear Flight Test Trajectory Controllers for Aircraft", *Journal of Guidance, Control, and Dynamics*, Vol. 10, Jan.-Feb. 1987, pp. 67-72.

[12] Lane, S. H., and Stengel, R. F., "Flight Control Design Using Nonlinear Inverse Dynamics", *Automatica*, Vol. 24, No. 4, 1988, pp. 471-483.

[13] Menon, P. K., Chatterji, G. B. and Cheng, V. H. L., "A Two-Time-Scale Autopilot for High Performance Aircraft," *AIAA Guidance, Navigation and Control Conference*, Aug. 12-14, 1991, New Orleans, LA.

[14] Menon, P. K., "Synthesis of Robust Nonlinear Autopilots Using Differential Game Theory," *American Control Conference*, June 26-28, 1991, Boston, MA.

[15] Menon, P. K., "Nonlinear Command Augmentation System for a High Performance Aircraft," *AIAA Guidance, Navigation and Control Conference*, August 9 -11, 1992, Monterey, CA.

[16] Menon, P. K., and Marduke, Y., "Design of Nonlinear Autopilots for High Angle of Attack Missiles," AIAA Guidance, Navigation and Control Conference, July 29 - 31, 1996, San Diego, CA.

[17] Menon, P. K., Iragavarapu, V. R., and Ohlmeyer, E. J., "Nonlinear Missile Autopilot Design Using Time-Scale Separation," *AIAA Guidance, Navigation and Control Conference*, August 11-13, 1997, New Orleans, LA.

[18] Menon, P. K., and Ohlmeyer, E. J., "Integrated Guidance-Control Systems for Fixed-Aim Warhead Missiles", *AIAA Missile Sciences Conference*, November 7-9, 2000, Monterey, CA.

[19] Menon, P. K. and Ohlmeyer, E. J., "Integrated Design of Agile Missile Guidance and Autopilot Systems", *IFAC Journal of Control Engineering Practice*, 2001, Vol. 9, pp. 1095-1106.

[20] Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, John Wiley, 1965, New York, NY.

[21] Kailath, T., *Linear Systems*, Prentice-Hall, 1980, Englewood Cliffs, NJ.

[22] Gerald, C. F., *Applied Numerical Analysis*, 1978, Addison-Wesley, Reading, MA.

[23] Brogan, W. L., *Modern Control Theory*, Prentice Hall, 1991, Upper Saddle River, NJ.

[24] Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, 1975, New York, NY.

[25] Menon, P. K., et al., Nonlinear Synthesis Tools™ for Use with MATLAB®, Optimal Synthesis Inc., 2003, Los Altos, CA.

[26] Menon, P. K., Iragavarapu, V. R., and Sweriduk, G. D., "Software Tools for Nonlinear Missile Autopilot Design," *AIAA Guidance, Navigation and Control Conference*, August 9-11, 1999, Portland, OR.

[27] Menon, P. K., Njaka, C. E., and Cheng, V. H. L., "Nonlinear Flight Control Using an Embedded Vehicle Computer Model," *AIAA Guidance, Navigation and Control Conference*, Aug. 7-9, 1995, Baltimore, MD.

[28] Njaka, C. E., Menon, P. K., and Cheng, V. H. L., "Towards an Advanced Nonlinear Rotorcraft Flight Control System Design," *Digital Avionics Systems Conference*, Oct. 31-Nov. 3, 1994, Phoenix, AZ.

[29] Cheng, V. H. L., Njaka, C. E., and Menon, P. K., "Practical Design Methodologies for Robust Nonlinear Flight Control," *AIAA Guidance, Navigation and Control Conference*, July 29 - 31, 1996, San Diego, CA.

[30] Menon, P. K., Lam, T., Crawford, L. S., and Cheng, V. H. L., "Real-Time Computational Methods for SDRE Nonlinear Control of Missiles", *American Control Conference*, May 8 –10, 2002, Anchorage, AK.